WHAT IS CLAIMED IS:

1. A cache comprising:

a cache line; and

an indicator associated with a unit-sized portion of said cache line,

wherein said indicator indicates whether said unit-sized portion is accessed.

2. The cache of claim 1, wherein said unit-sized portion is a byte.

3. The cache of claim 1, wherein said indicator further indicates a number of times that said unit-sized portion is accessed.

4. The cache of claim 1, wherein said cache is employed in a system comprising a processor for:

executing code;

evaluating said indicator, wherein said indicator indicates whether a unit of data in a memory is accessed during said execution; and

compiling said code to place said unit in a line of said memory if said unit is accessed during said execution,

wherein said line of said memory is designated to contain, in contiguous locations, a plurality of units of said data that are accessed during said execution.

5. The cache of claim 1, wherein said cache is employed in a system comprising a processor for:

determining whether a unit of data in a memory is likely to be accessed during an execution of code; and

compiling said code to place said unit in a line of said memory if said unit is likely to be accessed during said execution,

wherein said line of said memory is designated to contain, in contiguous locations, a plurality of units of said data that are likely to be accessed during said execution.

6. The cache of claim 5, wherein said determining comprises:
executing said code in a training session; and
evaluating said indicator to determine whether said unit is accessed during said training session.

7. The cache of claim 6, wherein said determining further comprises:
initializing said indicator;
repeating said evaluating after continuing said training session for an interval of time; and
determining an average rate of access of said unit during said training session.

8. The cache of claim 6, wherein said determining further comprises:
initializing said indicator;
repeating said evaluating after continuing said training session for an interval of time; and
determining a statistical ranking of a usage of said unit with respect to a usage of other units of said data during said training session.

9. The cache of claim 1,
wherein said unit-sized portion is a byte, and
wherein said cache is employed in a system comprising a processor for:
executing code in a training session;
evaluating said indicator, wherein said indicator indicates whether a byte of data in a memory is accessed during said training session; and

compiling said code to place said byte in a line of said memory if said byte is accessed during said training session,

wherein said line of said memory is designated to contain, in contiguous locations, a plurality of bytes of said data that are accessed during said training session.

10. A method for determining an arrangement of data in a memory for efficient operation of a cache, said method comprising:

determining whether a unit of said data is accessed during an execution of code; and

compiling said code to place said unit in a line of said memory if said unit is accessed during said execution,

wherein said line of said memory is designated to contain, in contiguous locations, a plurality of units of said data that are accessed during said execution.

11. The method of claim 10, wherein said unit is a byte-sized portion of said data.

12. The method of claim 10, wherein said determining comprises:

executing said code in a training session;

evaluating an indicator that is associated with a unit-sized portion of a line of said cache into which said unit is cached,

wherein said indicator indicates whether said unit is accessed during said training session.

13. A method for determining an arrangement of data in a memory for efficient operation of a cache, said method comprising:

determining whether a unit of said data is likely to be accessed during an execution of code; and

compiling said code to place said unit in a line of said memory if said unit is likely to be accessed during said execution,

wherein said line of said memory is designated to contain, in contiguous locations, a plurality of units of said data that are likely to be accessed during said execution.

14. The method of claim 13, wherein said unit is a byte-sized portion of said data.

15. The method of claim 13, wherein said determining comprises:

executing said code during a training session;

evaluating an indicator that is associated with a unit-sized portion of a line of said cache into which said unit is cached during said training session,

wherein said indicator indicates whether said unit is accessed during said training session.

16. The method of claim 15, wherein said determining further comprises:

initializing said indicator;

repeating said evaluating after continuing said training session for an interval of time; and

determining an average rate of access of said unit during said training session.

17. The method of claim 15, wherein said determining further comprises:

initializing said indicator;

repeating said evaluating after continuing said training session for an interval of time; and

determining a statistical ranking of a usage of said unit with respect to a usage of other units of said data during said training session.

18. A method for determining an arrangement of data in a memory for efficient operation of a cache, said method comprising:

executing code during a training session;

determining whether a byte of said data is accessed during said training session; and

compiling said code to place said byte in a line of said memory if said byte is accessed during said training session,

wherein said determining evaluates an indicator that is associated with a byte-sized portion of a line of said cache into which said byte is cached,

wherein said indicator indicates whether said byte is accessed during said training session,

wherein said line of said memory is designated to contain, in contiguous locations, a plurality of bytes of said data that are accessed during said training session.

19. A storage media that contains instructions for controlling a processor to, in turn, perform a method for determining an arrangement of data in a memory for efficient operation of a cache, said storage media comprising:

instructions for controlling said processor to determine whether a unit of said data is accessed during an execution of code; and

instructions for controlling said processor to compile said code to place said unit in a line of said memory if said unit is accessed during said execution,

wherein said line of said memory is designated to contain, in contiguous locations, a plurality of units of said data that are accessed during said execution.